Optimal Algorithm for the Contextual Bandit problem

Alekh Agarwal[†] Daniel Hsu[‡] Satyen Kale[♯]
John Langford[†] Lihong Li[†] Rob Schapire[†]

[†]Microsoft Research, [‡]Columbia University,
[♯]Google Research, [⋆]Princeton University

1. Introduction

Learning to interact: example #1

Loop:

1. Patient arrives with symptoms, medical history, genome ...
2. Physician prescribes treatment.
3. Patient's health responds (*e.g.*, improves, worsens).

Goal: prescribe treatments that yield good health outcomes.

Learning to interact: example #2

Loop:

1. User visits website with profile, browsing history . . .
2. Website operator chooses content/ads to display.
3. User reacts to content/ads (*e.g.*, click, “like”).

Goal: choose content/ads that yield desired user behavior.

Contextual bandit setting (i.i.d. version)

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context x_t.
2. Choose action $a_t \in \mathcal{A}$.
3. Collect reward $r_t(a_t)$.

Task: Design an algorithm for choosing a_t's that yield high reward.

Contextual bandit setting (i.i.d. version)

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context x_t.
2. Choose action $a_t \in \mathcal{A}$.
3. Collect reward $r_t(a_t)$.

Task: Design an algorithm for choosing a_t's that yield high reward.

Contextual setting: use features x_t to choose good actions a_t.

Contextual bandit setting (i.i.d. version)

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context x_t.
2. Choose action $a_t \in \mathcal{A}$.
3. Collect reward $r_t(a_t)$.

Task: Design an algorithm for choosing a_t's that yield high reward.

Contextual setting: use features x_t to choose good actions a_t.
Bandit setting: $r_t(a)$ for $a \neq a_t$ is not observed.
$\Longrightarrow$ Exploration vs. exploitation dilemma

(cf. non-bandit setting: whole reward vector $\boldsymbol{r}_t \in [0,1]^{\mathcal{A}}$ observed.)

Learning objective in the contextual bandit setting

No single action is good in all situations: **must exploit context**.

Learning objective in the contextual bandit setting

No single action is good in all situations: **must exploit context**.

Policy class Π: set of functions ("policies") $\pi : \mathcal{X} \to \mathcal{A}$
(*e.g.*, advice of experts, linear classifiers, neural networks).

Learning objective in the contextual bandit setting

No single action is good in all situations: **must exploit context**.

Policy class Π: set of functions ("policies") $\pi : \mathcal{X} \rightarrow \mathcal{A}$
(*e.g.*, advice of experts, linear classifiers, neural networks).

Regret (*i.e.*, relative performance) to a policy class Π:

$$\underbrace{\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t))}_{\text{total reward of best policy}} \quad - \quad \underbrace{\sum_{t=1}^{T} r_t(a_t)}_{\text{total reward of learner}}$$

... a strong benchmark when Π contains a policy with high reward.

Learning objective in the contextual bandit setting

No single action is good in all situations: **must exploit context**.

Policy class Π: set of functions ("policies") $\pi : \mathcal{X} \to \mathcal{A}$
(*e.g.*, advice of experts, linear classifiers, neural networks).

Regret (*i.e.*, relative performance) to a policy class Π:

$$\underbrace{\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t))}_{\text{total reward of best policy}} \quad - \quad \underbrace{\sum_{t=1}^{T} r_t(a_t)}_{\text{total reward of learner}}$$

. . . a strong benchmark when Π contains a policy with high reward.

Regret is sublinear (in T) $\implies$ (Avg.) per-round regret $\to 0$.

Challenge #1: computation

Feedback that learner observes: reward of chosen action $r_t(a_t)$
$\longrightarrow$ only directly relevant to $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.

Challenge #1: computation

Feedback that learner observes: reward of chosen action $r_t(a_t)$
$\longrightarrow$ only directly relevant to $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.

Separate explicit bookkeeping for each policy $\pi \in \Pi$ becomes **computationally intractable** when Π is large (or infinite!).

Arg max oracle (AMO): supervised learning

In many cases, we know how to exploit structure of Π to design efficient algorithms or heuristics!

Arg max oracle (AMO): supervised learning

In many cases, we know how to exploit structure of Π to design efficient algorithms or heuristics!

Given **fully labeled** data $(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times [0,1]^{\mathcal{A}}$, the AMO returns

$$\arg\max_{\pi \in \Pi} \sum_{i=1}^{t} \rho_i(\pi(x_i)).$$

Arg max oracle (AMO): supervised learning

In many cases, we know how to exploit structure of Π to design efficient algorithms or heuristics!

Given **fully labeled** data $(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times [0,1]^{\mathcal{A}}$, the AMO returns

$$\arg\max_{\pi \in \Pi} \sum_{i=1}^{t} \rho_i(\pi(x_i)).$$

AMO is an abstraction for efficient search through Π.

Arg max oracle (AMO): supervised learning

In many cases, we know how to exploit structure of Π to design efficient algorithms or heuristics!

Given **fully labeled** data $(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times [0,1]^{\mathcal{A}}$, the AMO returns

$$\arg\max_{\pi \in \Pi} \sum_{i=1}^{t} \rho_i(\pi(x_i)).$$

AMO is an abstraction for efficient search through Π.

In practice: implement using standard heuristics—e.g., convex relaxations, backpropagation—for cost-sensitive multi-class learning.

Arg max oracle (AMO): supervised learning

In many cases, we know how to exploit structure of Π to design efficient algorithms or heuristics!

Given **fully labeled** data $(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times [0,1]^{\mathcal{A}}$, the AMO returns

$$\arg\max_{\pi \in \Pi} \sum_{i=1}^{t} \rho_i(\pi(x_i)).$$

AMO is an abstraction for efficient search through Π.

In practice: implement using standard heuristics—e.g., convex relaxations, backpropagation—for cost-sensitive multi-class learning.

But requires **complete reward vectors** $\boldsymbol{\rho}_i$; not directly usable for contextual bandits.

Challenge #2: exploration

Possible approach: AMO + **simple random exploration**

1: In first T_0 rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get unbiased estimates $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \in [T_0]$.
2: Get $\tilde{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t \in [T_0]})$.
3: Use $a_t := \tilde{\pi}(x_t)$ in round $t > T_0$.

Challenge #2: exploration

Possible approach: AMO + **simple random exploration**

1: In first T_0 rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get unbiased estimates $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \in [T_0]$.
2: Get $\tilde{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t \in [T_0]})$.
3: Use $a_t := \tilde{\pi}(x_t)$ in round $t > T_0$.

But $$\mathbb{E}_{(x,\boldsymbol{r})}[r(\tilde{\pi}(x))] \approx \max_{\pi \in \Pi} \mathbb{E}_{(x,\boldsymbol{r})}[r(\pi(x))] - \Omega\left(\frac{1}{\sqrt{T_0}}\right)$$

(Dependencies on $|\mathcal{A}|$ and $|\Pi|$ hidden.)

Challenge #2: exploration

Possible approach: AMO + **simple random exploration**

1: In first T_0 rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get unbiased estimates $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \in [T_0]$.
2: Get $\tilde{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t \in [T_0]})$.
3: Use $a_t := \tilde{\pi}(x_t)$ in round $t > T_0$.

But $\mathbb{E}_{(x,\boldsymbol{r})}[r(\tilde{\pi}(x))] \approx \max_{\pi \in \Pi} \mathbb{E}_{(x,\boldsymbol{r})}[r(\pi(x))] - \Omega\left(\frac{1}{\sqrt{T_0}}\right)$

...so regret with this approach (with best T_0) could be as large as

$$\Omega\left(T_0 + \frac{1}{\sqrt{T_0}}(T - T_0)\right)$$

(Dependencies on $|\mathcal{A}|$ and $|\Pi|$ hidden.)

Challenge #2: exploration

Possible approach: AMO + **simple random exploration**

1: In first T_0 rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get unbiased estimates $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \in [T_0]$.
2: Get $\tilde{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t \in [T_0]})$.
3: Use $a_t := \tilde{\pi}(x_t)$ in round $t > T_0$.

But $\mathbb{E}_{(x,\boldsymbol{r})}[r(\tilde{\pi}(x))] \approx \max_{\pi \in \Pi} \mathbb{E}_{(x,\boldsymbol{r})}[r(\pi(x))] - \Omega\left(\frac{1}{\sqrt{T_0}}\right)$

...so regret with this approach (with best T_0) could be as large as

$$\Omega\left(T_0 + \frac{1}{\sqrt{T_0}}(T - T_0)\right) \sim T^{2/3} \gg T^{1/2}.$$

(Dependencies on $|\mathcal{A}|$ and $|\Pi|$ hidden.)

Algorithms for contextual bandits

Let $K := |\mathcal{A}|$ and $N := |\Pi|$.

Our result [AHKLLS'14]: a new, fast and simple algorithm.
Optimal regret bound $\tilde{O}(\sqrt{KT \log N})$.
$\tilde{O}(\sqrt{TK})$ calls to AMO overall.

Algorithms for contextual bandits

Let $K := |\mathcal{A}|$ and $N := |\Pi|$.

Our result [AHKLLS'14]: a new, fast and simple algorithm.
Optimal regret bound $\tilde{O}(\sqrt{KT \log N})$.
$\tilde{O}(\sqrt{TK})$ calls to AMO overall.

Previous work:

[ACBFS'02] Exp4 algorithm (exponential weights).
Optimal regret bound $O(\sqrt{KT \log N})$.
Requires explicit enumeration of Π in every round.

Algorithms for contextual bandits

Let $K := |\mathcal{A}|$ and $N := |\Pi|$.

Our result [AHKLLS'14]: a new, fast and simple algorithm.
Optimal regret bound $\tilde{O}(\sqrt{KT \log N})$.
$\tilde{O}(\sqrt{TK})$ calls to AMO overall.

Previous work:

[ACBFS'02] Exp4 algorithm (exponential weights).
Optimal regret bound $O(\sqrt{KT \log N})$.
Requires explicit enumeration of Π in every round.

[LZ'07] ϵ-greedy variant (uniform exploration).
Suboptimal regret bound $\tilde{O}(T^{2/3}(K \log N)^{1/3})$.
One call to AMO overall.

Algorithms for contextual bandits

Let $K := |\mathcal{A}|$ and $N := |\Pi|$.

Our result [AHKLLS'14]: a new, fast and simple algorithm.
Optimal regret bound $\tilde{O}(\sqrt{KT \log N})$.
$\tilde{O}(\sqrt{TK})$ calls to AMO overall.

Previous work:

[ACBFS'02] Exp4 algorithm (exponential weights).
Optimal regret bound $O(\sqrt{KT \log N})$.
Requires explicit enumeration of Π in every round.

[LZ'07] ϵ-greedy variant (uniform exploration).
Suboptimal regret bound $\tilde{O}(T^{2/3}(K \log N)^{1/3})$.
One call to AMO overall.

[DHKKLRZ'11] "efficient" algorithm (careful exploration).
Optimal regret bound $\tilde{O}(\sqrt{KT \log N})$.
$O(T^6 K^4)$ calls to AMO overall.

Rest of the talk

Components of the new algorithm: Importance-weighted LOw-Variance Epoch-Timed Oracleized CONtextual BANDITS

1. "Classical" tricks: randomization, inverse probability weighting.
2. Efficient algorithm for balancing exploration/exploitation.
3. Additional tricks: warm-start and epoch structure.

Note: we assume $(x_t, \boldsymbol{r}_t)$ i.i.d. from $\mathcal{D}$
(whereas Exp4 also works in adversarial setting).

Outline

1. Introduction
2. Classical tricks
3. Construction of good policy distributions
4. Additional tricks: warm-start and epoch structure

2. Classical tricks

What would've happened if I had done X?

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context x_t.
2. Choose action $a_t \in \mathcal{A}$.
3. Collect reward $r_t(a_t)$.

What would've happened if I had done X?

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context x_t.
2. Choose action $a_t \in \mathcal{A}$.
3. Collect reward $r_t(a_t)$.

Q: How do I learn about $r_t(a)$ for actions a I don't actually take?

What would've happened if I had done X?

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context x_t.
2. Choose action $a_t \in \mathcal{A}$.
3. Collect reward $r_t(a_t)$.

Q: How do I learn about $r_t(a)$ for actions a I don't actually take?

A: *Randomize*. Draw $a_t \sim \boldsymbol{p}_t$ for some pre-specified prob. dist. $\boldsymbol{p}_t$.

Inverse probability weighting

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}\,. \quad \hat{r}_t(a) \;:=\; \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)}$$

Inverse probability weighting

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A} . \quad \hat{r}_t(a) := \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} = \begin{cases} \frac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}$$

Inverse probability weighting

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}\,. \quad \hat{r}_t(a) \;:=\; \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} \;=\; \begin{cases} \frac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] \;=\; \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} \;=\; r_t(a).$$

Inverse probability weighting

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}\,. \quad \hat{r}_t(a) \;:=\; \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} \;=\; \begin{cases} \frac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] \;=\; \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} \;=\; r_t(a).$$

Range and variance: upper-bounded by $1/p_t(a)$.

Inverse probability weighting

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A} \, . \quad \hat{r}_t(a) \ := \ \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} \ = \ \begin{cases} \frac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] \ = \ \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} \ = \ r_t(a).$$

Range and variance: upper-bounded by $1/p_t(a)$.

Expected reward of policy: $\mathrm{Rew}(\pi) = \mathbb{E}_{(x,\boldsymbol{r})}[r(\pi(x)]$

Unbiased estimator of total reward: $\widehat{\mathrm{Rew}}_t(\pi) := \sum_{i=1}^{t} \hat{r}_i(\pi(x_i))$.

Inverse probability weighting

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A} \,\blacksquare \quad \hat{r}_t(a) := \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} = \begin{cases} \frac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}[\hat{r}_t(a)] = \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} = r_t(a).$$

Range and variance: upper-bounded by $1/p_t(a)$.

Expected reward of policy: $\text{Rew}(\pi) = \mathbb{E}_{(x,\boldsymbol{r})}[r(\pi(x)]$

Unbiased estimator of total reward: $\widehat{\text{Rew}}_t(\pi) := \sum_{i=1}^{t} \hat{r}_i(\pi(x_i))$.

How should we choose the $\boldsymbol{p}_t$**?**

Hedging over policies

Get action distributions via policy distributions.

$$\underbrace{(\boldsymbol{W}, x)}_{\text{(policy distribution, context)}} \quad \mapsto \quad \underbrace{\boldsymbol{p}}_{\text{action distribution}}$$

Hedging over policies

Get action distributions via policy distributions.

$$\underbrace{(\boldsymbol{W}, x)}_{\text{(policy distribution, context)}} \quad \mapsto \quad \underbrace{\boldsymbol{p}}_{\text{action distribution}}$$

Policy distribution: $\boldsymbol{W} = (W(\pi) : \pi \in \Pi)$
probability dist. over policies π in the policy class Π

Hedging over policies

Get action distributions via policy distributions.

$$\underbrace{(\boldsymbol{W}, x)}_{\text{(policy distribution, context)}} \quad \mapsto \quad \underbrace{\boldsymbol{p}}_{\text{action distribution}}$$

Pick initial distribution $\boldsymbol{W}_1$ over policies Π.
for round $t = 1, 2, \ldots$ **do**
  Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
  Observe context x_t.
  Compute distribution $\boldsymbol{p}_t$ over $\mathcal{A}$ (using $\boldsymbol{W}_t$ and x_t).
  Pick action $a_t \sim \boldsymbol{p}_t$.
  Collect reward $r_t(a_t)$.
  Compute new distribution $\boldsymbol{W}_{t+1}$ over policies Π.
end for

Projections of policy distributions

Given policy distribution $\boldsymbol{W}$ and context x,

$$\forall a \in \mathcal{A} \,.\quad W(a|x) := \sum_{\pi \in \Pi} W(\pi) \cdot \mathbb{1}\{\pi(x) = a\}$$

(so $\boldsymbol{W} \mapsto \boldsymbol{W}(\cdot|x)$ is a linear map).

Projections of policy distributions

Given policy distribution $\boldsymbol{W}$ and context x,

$$\forall a \in \mathcal{A}\,. \quad W(a|x) := \sum_{\pi \in \Pi} W(\pi) \cdot \mathbb{1}\{\pi(x) = a\}$$

(so $\boldsymbol{W} \mapsto \boldsymbol{W}(\cdot|x)$ is a linear map).

We actually use

$$\boldsymbol{p}_t := \boldsymbol{W}_t^{\mu_t}(\,\cdot\,|x_t) := (1 - K\mu_t)\boldsymbol{W}_t(\,\cdot\,|x_t) + \mu_t$$

so every action has probability at least μ_t (*to be determined*).

Basic algorithm structure

```
Pick initial distribution W_1 over policies Π.
for round t = 1, 2, ... do
    Nature draws (x_t, r_t) from dist. D over X × [0,1]^A.
    Observe context x_t.
    Compute action distribution p_t := W_t^{μ_t}( · | x_t).
    Pick action a_t ∼ p_t.
    Collect reward r_t(a_t).
    Compute new distribution W_{t+1} over policies Π.
end for
```

Q: How do we choose $\boldsymbol{W}_t$ for good exploration/exploitation?

Basic algorithm structure

```
Pick initial distribution W_1 over policies Π.
for round t = 1, 2, ... do
    Nature draws (x_t, r_t) from dist. D over X × [0,1]^A.
    Observe context x_t.
    Compute action distribution p_t := W_t^{μ_t}( · | x_t).
    Pick action a_t ~ p_t.
    Collect reward r_t(a_t).
    Compute new distribution W_{t+1} over policies Π.
end for
```

Q: How do we choose $\boldsymbol{W}_t$ for good exploration/exploitation?

Caveat: $\boldsymbol{W}_t$ must be efficiently computable + representable!

3. Construction of good policy distributions

Our approach

- Define convex feasibility problem (over distributions $\boldsymbol{W}$ on Π) such that solutions yield optimal regret bounds.

Our approach

- Define convex feasibility problem (over distributions $\boldsymbol{W}$ on Π) such that solutions yield optimal regret bounds.
- Design algorithm that finds a *sparse* solution $\boldsymbol{W}$.

Our approach

- Define convex feasibility problem (over distributions $\boldsymbol{W}$ on Π) such that solutions yield optimal regret bounds.
- Design algorithm that finds a *sparse* solution $\boldsymbol{W}$.

  Algorithm only accesses Π via calls to AMO
  $\Longrightarrow \text{nnz}(\boldsymbol{W}) = \#$ calls to AMO

An optimal but inefficient algorithm

Policy_Elimination

Let $\Pi_1 = \Pi$.

An optimal but inefficient algorithm

Policy_Elimination

Let $\Pi_1 = \Pi$. For each $t = 1, 2, \ldots$:

1. Choose distribution W_t over Π_t such that

$$\forall \pi \in \Pi_t : \quad \mathbb{E}_x \left[\frac{1}{W_t(\pi(x)|x)} \right] \leq K$$

An optimal but inefficient algorithm

Policy_Elimination

Let $\Pi_1 = \Pi$. For each $t = 1, 2, \ldots$:

1. Choose distribution W_t over Π_t such that

$$\forall \pi \in \Pi_t : \quad \mathop{\mathbb{E}}_{x}\left[\frac{1}{W_t(\pi(x)|x)}\right] \leq K$$

2. Let $\overline{\mathrm{Rew}}_t(\pi) = \frac{1}{t}\widehat{\mathrm{Rew}}_t(\pi)$, i.e. the average of all the estimators for $\mathrm{Rew}(\pi)$ so far. Let

$$\Pi_{t+1} = \left\{\pi \in \Pi_t : \overline{\mathrm{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\mathrm{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right)\right\}$$

Analysis Sketch: Distribution Selection Step

Choose distribution W_t over Π_t such that

$$\forall \pi \in \Pi_t : \underset{x}{\mathbb{E}}\left[\frac{1}{W_t(\pi(x)|x)}\right] \le K$$

Analysis Sketch: Distribution Selection Step

Choose distribution W_t over Π_t such that

$$\forall \pi \in \Pi_t : \ \mathop{\mathbb{E}}_{x}\left[\frac{1}{W_t(\pi(x)|x)}\right] \leq K$$

- Ensures that $\forall \pi \in \Pi_t$:

$$\mathop{\mathbf{Var}}_{x}[\hat{r}_t(\pi(x_t))] \ \leq \ O(1).$$

Analysis Sketch: Distribution Selection Step

Choose distribution W_t over Π_t such that

$$\forall \pi \in \Pi_t : \mathop{\mathbb{E}}_{x}\left[\frac{1}{W_t(\pi(x)|x)}\right] \leq K$$

- Ensures that $\forall \pi \in \Pi_t$:

$$\mathop{\mathbf{Var}}_{x}[\hat{r}_t(\pi(x_t))] \;\leq\; O(1).$$

- Hence, averaging over t iterations, we have $\forall \pi \in \Pi_t$:

$$\mathop{\mathbf{Var}}_{x}[\overline{\mathrm{Rew}}_t(\pi)] \;\leq\; O\left(\frac{1}{t}\right).$$

Analysis Sketch: Distribution Selection Step

Choose distribution W_t over Π_t such that

$$\forall \pi \in \Pi_t : \mathop{\mathbb{E}}_x \left[\frac{1}{W_t(\pi(x)|x)} \right] \leq K$$

- Ensures that $\forall \pi \in \Pi_t$:

$$\mathop{\mathbf{Var}}_x [\hat{r}_t(\pi(x_t))] \ \leq \ O(1).$$

- Hence, averaging over t iterations, we have $\forall \pi \in \Pi_t$:

$$\mathop{\mathbf{Var}}_x [\overline{\mathrm{Rew}}_t(\pi)] \ \leq \ O\left(\frac{1}{t}\right).$$

- Martingale concentration bounds imply that w.h.p. $\forall \pi \in \Pi_t$:

$$|\overline{\mathrm{Rew}}_t(\pi) - \mathrm{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

Analysis Sketch: Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\mathsf{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\mathsf{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

Analysis Sketch: Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\mathsf{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\mathsf{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

- W.h.p. $\forall \pi \in \Pi_t$:

$$|\overline{\mathsf{Rew}}_t(\pi) - \mathsf{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

Analysis Sketch: Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\mathrm{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\mathrm{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

- W.h.p. $\forall \pi \in \Pi_t$:

$$|\overline{\mathrm{Rew}}_t(\pi) - \mathrm{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

- W.h.p. $\forall \pi \in \Pi_{t+1}$:

$$|\,\mathrm{Rew}(\pi^\star) - \mathrm{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

Analysis Sketch: Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\text{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\text{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

- W.h.p. $\forall \pi \in \Pi_t$:

$$|\overline{\text{Rew}}_t(\pi) - \text{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

- W.h.p. $\forall \pi \in \Pi_{t+1}$:

$$|\text{Rew}(\pi^\star) - \text{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

- Thus, expected regret in time $t + 1$ is $O(\frac{1}{\sqrt{t}})$.

Analysis Sketch: Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\text{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\text{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

- W.h.p. $\forall \pi \in \Pi_t$:

$$|\overline{\text{Rew}}_t(\pi) - \text{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

- W.h.p. $\forall \pi \in \Pi_{t+1}$:

$$|\,\text{Rew}(\pi^\star) - \text{Rew}(\pi)| \leq O\left(\frac{1}{\sqrt{t}}\right).$$

- Thus, expected regret in time $t+1$ is $O(\frac{1}{\sqrt{t}})$.
- Thus, total regret is $\sum_{t=1}^{T} O(\frac{1}{\sqrt{t}}) = O(\sqrt{T})$.

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

Why should such a distribution exist?

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

Why should such a distribution exist?

Answer: Minimax magic.

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

Why should such a distribution exist?

Answer: Minimax magic.

$$\min_W \max_\pi \mathop{\mathbb{E}}_x \left[\frac{1}{W(\pi(x)|x)} \right] = \min_W \max_U \mathop{\mathbb{E}}_{x, \pi \sim U} \left[\frac{1}{W(\pi(x)|x)} \right]$$

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

Why should such a distribution exist?

Answer: Minimax magic.

$$\min_W \max_\pi \mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] = \min_W \max_U \mathbb{E}_{x,\pi\sim U} \left[\frac{1}{W(\pi(x)|x)} \right]$$
$$= \max_U \min_W \mathbb{E}_{x,\pi\sim U} \left[\frac{1}{W(\pi(x)|x)} \right]$$

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)}\right] \leq K$.

Why should such a distribution exist?

Answer: Minimax magic.

$$\begin{aligned}
\min_W \max_\pi \mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)}\right] &= \min_W \max_U \mathop{\mathbb{E}}_{x,\pi\sim U} \left[\frac{1}{W(\pi(x)|x)}\right] \\
&= \max_U \min_W \mathop{\mathbb{E}}_{x,\pi\sim U} \left[\frac{1}{W(\pi(x)|x)}\right] \\
&\leq \max_U \mathop{\mathbb{E}}_{x,\pi\sim U} \left[\frac{1}{U(\pi(x)|x)}\right]
\end{aligned}$$

Existence of Distribution

Key step: Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

Why should such a distribution exist?

Answer: Minimax magic.

$$
\begin{aligned}
\min_W \max_\pi \mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] &= \min_W \max_U \mathbb{E}_{x,\pi \sim U} \left[\frac{1}{W(\pi(x)|x)} \right] \\
&= \max_U \min_W \mathbb{E}_{x,\pi \sim U} \left[\frac{1}{W(\pi(x)|x)} \right] \\
&\leq \max_U \mathbb{E}_{x,\pi \sim U} \left[\frac{1}{U(\pi(x)|x)} \right] \\
&= \max_U \mathbb{E}_x \left[\sum_{a \in [K]} \frac{U(a|x)}{U(a|x)} \right] \leq K.
\end{aligned}
$$

Problems with the algorithm

Distribution Selection Step

Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \le K$.

Problems with the algorithm

Distribution Selection Step

Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

- Computing P is a convex optimization problem and takes poly(N) time.

Problems with the algorithm

Distribution Selection Step

Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

- Computing P is a convex optimization problem and takes poly(N) time.
- Computing P requires knowledge of actual data distribution.

Problems with the algorithm

Distribution Selection Step

Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \le K$.

- Computing P is a convex optimization problem and takes poly(N) time.
- Computing P requires knowledge of actual data distribution.

Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\text{Rew}}_t(\pi) \ge \max_{\pi' \in \Pi_t} \overline{\text{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

Problems with the algorithm

Distribution Selection Step

Choose W s.t. $\forall \pi \in \Pi_t$, we have $\mathbb{E}_x \left[\frac{1}{W(\pi(x)|x)} \right] \leq K$.

- Computing P is a convex optimization problem and takes poly(N) time.
- Computing P requires knowledge of actual data distribution.

Policy Elimination Step

$$\Pi_{t+1} = \left\{ \pi \in \Pi_t : \overline{\text{Rew}}_t(\pi) \geq \max_{\pi' \in \Pi_t} \overline{\text{Rew}}_t(\pi') - \Theta\left(\frac{1}{\sqrt{t}}\right) \right\}$$

- Policy Elimination Step takes $\Omega(N)$ time.

Properties of a good policy distribution

Low Regret and Low Variance constraints on $\boldsymbol{W}$:

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\text{Reg}}_t(\pi) \;\le\; \sqrt{Kt \log N}, \qquad \text{(LR)}$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \le K\left(1 + \frac{\widehat{\text{Reg}}_t(\pi)}{\sqrt{Kt \log N}}\right) \quad \forall \pi \in \Pi \quad \text{(LV)}$$

$\widehat{\text{Reg}}_t(\pi) := \max_{\pi' \in \Pi} \widehat{\text{Rew}}_t(\pi') - \widehat{\text{Rew}}_t(\pi)$, $\quad \mu_t := \sqrt{\frac{\log N}{Kt}}$, $\quad H_t := (x_1, \ldots, x_t)$

Properties of a good policy distribution

Low Regret and Low Variance constraints on $\boldsymbol{W}$:

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\leq\; \sqrt{Kt \log N}, \qquad \text{(LR)}$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \leq K\left(1 + \frac{\widehat{\mathrm{Reg}}_t(\pi)}{\sqrt{Kt \log N}}\right) \quad \forall \pi \in \Pi \quad \text{(LV)}$$

$\widehat{\mathrm{Reg}}_t(\pi) := \max_{\pi' \in \Pi} \widehat{\mathrm{Rew}}_t(\pi') - \widehat{\mathrm{Rew}}_t(\pi)$, $\quad \mu_t := \sqrt{\frac{\log N}{Kt}}$, $\quad H_t := (x_1, \ldots, x_t)$

Intuition: Allow higher variance for policies π with larger regret, as they should have low weight anyway.

Properties of a good policy distribution

Low **R**egret and **L**ow **V**ariance constraints on $\boldsymbol{W}$:

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\le\; Kt \cdot \mu_t, \tag{LR}$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \le K\left(1 + \frac{\widehat{\mathrm{Reg}}_t(\pi)}{Kt \cdot \mu_t}\right) \quad \forall \pi \in \Pi \tag{LV}$$

$$\widehat{\mathrm{Reg}}_t(\pi) := \max_{\pi' \in \Pi} \widehat{\mathrm{Rew}}_t(\pi') - \widehat{\mathrm{Rew}}_t(\pi), \quad \mu_t := \sqrt{\frac{\log N}{Kt}}, \quad H_t := (x_1, \ldots, x_t)$$

Intuition: Allow higher variance for policies π with larger regret, as they should have low weight anyway.

Properties of a good policy distribution

<u>L</u>ow <u>R</u>egret and <u>L</u>ow <u>V</u>ariance constraints on $\boldsymbol{W}$:

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\text{Reg}}_t(\pi) \;\leq\; Kt \cdot \mu_t, \qquad \text{(LR)}$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \leq K\left(1 + \frac{\widehat{\text{Reg}}_t(\pi)}{Kt \cdot \mu_t}\right) \quad \forall \pi \in \Pi \qquad \text{(LV)}$$

$$\widehat{\text{Reg}}_t(\pi) := \max_{\pi' \in \Pi} \widehat{\text{Rew}}_t(\pi') - \widehat{\text{Rew}}_t(\pi), \quad \mu_t := \sqrt{\frac{\log N}{Kt}}, \quad H_t := (x_1, \ldots, x_t)$$

$$\text{(LV)} \quad \Longrightarrow \quad \text{Reg}(\pi) \;\leq\; O\left(\widehat{\text{Reg}}_t(\pi) + Kt \cdot \mu_t\right) \quad \forall \pi \in \Pi;$$

$$\text{(LR,LV)} \quad \Longrightarrow \quad \sum_{\pi \in \Pi} W_t(\pi) \cdot \text{Reg}(\pi) \;\leq\; O(Kt \cdot \mu_t).$$

Properties of a good policy distribution

Low Regret and Low Variance constraints on $\boldsymbol{W}$:

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\leq\; Kt \cdot \mu_t, \tag{LR}$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \leq K\left(1 + \frac{\widehat{\mathrm{Reg}}_t(\pi)}{Kt \cdot \mu_t}\right) \quad \forall \pi \in \Pi \tag{LV}$$

$$\widehat{\mathrm{Reg}}_t(\pi) := \max_{\pi' \in \Pi} \widehat{\mathrm{Rew}}_t(\pi') - \widehat{\mathrm{Rew}}_t(\pi), \quad \mu_t := \sqrt{\frac{\log N}{Kt}}, \quad H_t := (x_1, \ldots, x_t)$$

Theorem: If we pick $\boldsymbol{W}_t$ satisfying (LR,LV) in every round t, then regret over all T rounds is $O\left(\sqrt{KT \log N}\right)$.

Properties of a good policy distribution

Low **R**egret and **L**ow **V**ariance constraints on $\boldsymbol{W}$:

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\leq\; Kt \cdot \mu_t, \qquad \text{(LR)}$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \leq K\left(1 + \frac{\widehat{\mathrm{Reg}}_t(\pi)}{Kt \cdot \mu_t}\right) \quad \forall \pi \in \Pi \qquad \text{(LV)}$$

$$\widehat{\mathrm{Reg}}_t(\pi) := \max_{\pi' \in \Pi} \widehat{\mathrm{Rew}}_t(\pi') - \widehat{\mathrm{Rew}}_t(\pi), \quad \mu_t := \sqrt{\frac{\log N}{Kt}}, \quad H_t := (x_1, \ldots, x_t)$$

Theorem: If we pick $\boldsymbol{W}_t$ satisfying (LR,LV) in every round t, then regret over all T rounds is $O\left(\sqrt{KT \log N}\right)$.

Critical question: Is it even feasible to satisfy (LR,LV)?

Minmax proof of feasibility (simplified)

$$\sum_{\pi \in \Pi} W(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\le\; \sqrt{Kt \log N},$$

$$\widehat{\mathbb{E}}_{x \in H_t}\left[\frac{1}{W(\pi(x)|x)}\right] \;\le\; K\left(1 + \frac{\widehat{\mathrm{Reg}}_t(\pi)}{\sqrt{Kt \log N}}\right) \quad \forall \pi \in \Pi$$

Minmax proof of feasibility (simplified)

$$\sum_{\pi \in \Pi} b(\pi) W(\pi) - 1 \ \leq \ 0,$$

$$\frac{1}{K} \widehat{\mathbb{E}}_{x \in H_t} \left[\frac{1}{W(\pi(x)|x)} \right] - (1 + b(\pi)) \ \leq \ 0 \ \ \forall \pi \in \Pi$$

$$b(\pi) := \widehat{\mathrm{Reg}}_t(\pi) / \sqrt{Kt \log N}$$

Minmax proof of feasibility (simplified)

$$\min_{\boldsymbol{W}\in\Delta^N} \max_{(U_o,\boldsymbol{U})\in\Delta^{N+1}} U_o\left(\sum_{\pi\in\Pi} b(\pi)W(\pi) - 1\right) + \sum_{\pi\in\Pi} U(\pi)\left(\frac{1}{K}\widehat{\mathbb{E}}_{x\in H_t}\left[\frac{1}{W(\pi(x)|x)}\right] - (1+b(\pi))\right) \le 0$$

$$b(\pi) := \widehat{\mathrm{Reg}}_t(\pi)/\sqrt{Kt\log N}$$

Minmax proof of feasibility (simplified)

$$\max_{(U_o,\boldsymbol{U})\in\Delta^{N+1}} \min_{\boldsymbol{W}\in\Delta^{N}} U_o\left(\sum_{\pi\in\Pi} b(\pi)W(\pi) - 1\right) + \sum_{\pi\in\Pi} U(\pi)\left(\frac{1}{K}\widehat{\mathbb{E}}_{x\in H_t}\left[\frac{1}{W(\pi(x)|x)}\right] - (1+b(\pi))\right) \le 0$$

$$b(\pi) := \widehat{\mathrm{Reg}}_t(\pi)/\sqrt{Kt\log N}$$

Minmax proof of feasibility (simplified)

$$\max_{(U_o,\boldsymbol{U})\in\Delta^{N+1}} \min_{\boldsymbol{W}\in\Delta^N} U_o\left(\sum_{\pi\in\Pi} b(\pi)W(\pi) - 1\right) + \sum_{\pi\in\Pi} U(\pi)\left(\frac{1}{K}\widehat{\mathbb{E}}_{x\in H_t}\left[\frac{1}{W(\pi(x)|x)}\right] - (1+b(\pi))\right) \leq 0$$

$$b(\pi) := \widehat{\text{Reg}}_t(\pi)/\sqrt{Kt\log N}$$

Choose $\boldsymbol{W} := \boldsymbol{U} + U_o\mathbf{1}^{\hat{\pi}}$ for $\hat{\pi} := \arg\min_{\pi\in\Pi} b(\pi)$
to verify that value of game ≤ 0.

Minmax proof of feasibility (simplified)

Choose $\boldsymbol{W} := \boldsymbol{U} + U_o \mathbf{1}^{\hat{\pi}}$ for $\hat{\pi} := \arg\min_{\pi \in \Pi} b(\pi)$ (so $b(\hat{\pi}) = 0$)
to verify that value of game ≤ 0.

$$\max_{(U_o, \boldsymbol{U}) \in \Delta^{N+1}} \min_{\boldsymbol{W} \in \Delta^N} U_o \left(\sum_{\pi \in \Pi} b(\pi) W(\pi) - 1 \right) + \sum_{\pi \in \Pi} U(\pi) \left(\frac{1}{K} \widehat{\mathbb{E}}_{x \in H_t} \left[\frac{1}{W(\pi(x)|x)} \right] - (1 + b(\pi)) \right)$$

Minmax proof of feasibility (simplified)

Choose $\boldsymbol{W} := \boldsymbol{U} + U_o \mathbf{1}^{\hat{\pi}}$ for $\hat{\pi} := \arg\min_{\pi \in \Pi} b(\pi)$ (so $b(\hat{\pi}) = 0$)
to verify that value of game ≤ 0.

$$\max_{(U_o, \boldsymbol{U}) \in \Delta^{N+1}} U_o \left(\sum_{\pi \in \Pi} b(\pi) U(\pi) - 1 \right) + \frac{1}{K} \widehat{\mathbb{E}}_{x \in H_t} \left[\sum_{a \in \mathcal{A}} \frac{U(a|x)}{W(a|x)} \right] - \sum_{\pi \in \Pi} U(\pi)(1 + b(\pi))$$

Minmax proof of feasibility (simplified)

Choose $\boldsymbol{W} := \boldsymbol{U} + U_o \mathbf{1}^{\hat{\pi}}$ for $\hat{\pi} := \arg\min_{\pi \in \Pi} b(\pi)$ (so $b(\hat{\pi}) = 0$)
to verify that value of game ≤ 0.

$$\max_{(U_o, \boldsymbol{U}) \in \Delta^{N+1}} (U_o - 1) \sum_{\pi \in \Pi} b(\pi) U(\pi) + \frac{1}{K} \widehat{\mathbb{E}}_{x \in H_t} \left[\sum_{a \in \mathcal{A}} \frac{U(a|x)}{W(a|x)} \right] - 1 \leq 0$$

Feasibility and sparsity

Feasibility of LR/LV constraints is implied by minimax argument.

Feasibility and sparsity

Feasibility of LR/LV constraints is implied by minimax argument.

"Monster" solution [DHKKLRZ'11]: Can solve (variant) of feasibility problem using Ellipsoid algorithm
(where separation oracle = AMO + Perceptron + another Ellipsoid).

Feasibility and sparsity

Feasibility of LR/LV constraints is implied by minimax argument.

"Monster" solution [DHKKLRZ'11]: Can solve (variant) of feasibility problem using Ellipsoid algorithm
(where separation oracle $=$ AMO $+$ Perceptron $+$ another Ellipsoid).

Existence of sparse(r) solution: given any (dense) solution, *probabilistic method* shows that there is an $\tilde{O}(\sqrt{Kt})$-sparse approximation with comparable LR and LV constraint bounds.

Feasibility and sparsity

Feasibility of LR/LV constraints is implied by minimax argument.

"Monster" solution [DHKKLRZ'11]: Can solve (variant) of feasibility problem using Ellipsoid algorithm
(where separation oracle = AMO + Perceptron + another Ellipsoid).

Existence of sparse(r) solution: given any (dense) solution, *probabilistic method* shows that there is an $\tilde{O}(\sqrt{Kt})$-sparse approximation with comparable LR and LV constraint bounds.

Efficient construction via "boosting"-type algorithm?

Coordinate descent algorithm

input Initial weights W.

loop
  If (LR) is violated, then replace W by cW.
  if there is a policy $\pi \in \Pi$ causing (LV) to be violated **then**
    set $W(\pi) := W(\pi) + \alpha$.
  else
    Halt and return W.
  end if
end loop

(Both $0 < c < 1$ and $\alpha > 0$ have closed form expressions.)

(Technical detail: actually optimize over subdistributions that may sum to < 1.)

Implementation via AMO

Checking violation of (LV) constraint: for all $\pi \in \Pi$,

$$\widehat{\mathbb{E}}_x\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \leq K\left(1 + \frac{\max_{\pi'} \widehat{\text{Rew}}_t(\pi') - \widehat{\text{Rew}}_t(\pi)}{Kt \cdot \mu_t}\right)$$

Implementation via AMO

Checking violation of (LV) constraint: for all $\pi \in \Pi$,

$$\frac{\widehat{\mathrm{Rew}}_t(\pi)}{t \cdot \mu_t} + \widehat{\mathbb{E}}_x\left[\frac{1}{W^{\mu_t}(\pi(x)|x)}\right] \leq K\left(1 + \frac{\max_{\pi'} \widehat{\mathrm{Rew}}_t(\pi')}{Kt \cdot \mu_t}\right)$$

Implementation via AMO

Checking violation of (LV) constraint: for all $\pi \in \Pi$,

$$\widehat{\mathsf{Rew}}_t(\pi) + t \cdot \widehat{\mathbb{E}}_x\left[\frac{\mu_t}{W^{\mu_t}(\pi(x)|x)}\right] \le Kt \cdot \mu_t + \max_{\pi'} \widehat{\mathsf{Rew}}_t(\pi')$$

Implementation via AMO

Checking violation of (LV) constraint: for all $\pi \in \Pi$,

$$\widehat{\mathrm{Rew}}_t(\pi) + t \cdot \widehat{\mathbb{E}}_x\left[\frac{\mu_t}{W^{\mu_t}(\pi(x)|x)}\right] \le Kt \cdot \mu_t + \max_{\pi'} \widehat{\mathrm{Rew}}_t(\pi')$$

1. Obtain $\hat{\pi} := \mathrm{AMO}((x_1, \hat{\boldsymbol{r}}_1), \ldots, (x_t, \hat{\boldsymbol{r}}_t))$.
2. Create fictitious rewards for each $i = 1, 2, \ldots, t$:

$$\tilde{r}_i(a) \; := \; \frac{\mu}{W^{\mu_t}(a|x_i)} + \hat{r}_i(a) \quad \forall a \in \mathcal{A}.$$

   Obtain $\tilde{\pi} := \mathrm{AMO}((x_1, \tilde{\boldsymbol{r}}_1), \ldots, (x_t, \tilde{\boldsymbol{r}}_t))$.
3. $\widetilde{\mathrm{Rew}}_t(\tilde{\pi}) > Kt \cdot \mu_t + \widehat{\mathrm{Rew}}_t(\hat{\pi})$ iff (LV) is violated by $\tilde{\pi}$.

Iteration bound for coordinate descent

Using unnormalized relative entropy-based potential function

$$\Phi(W) := t\mu_t\left(\frac{\widehat{\mathbb{E}}_{x\in H_t}\left[\mathrm{RE}(\mathrm{unif} \,\|\, W^{\mu_t}(\cdot|x))\right]}{1 - K\mu_t} + \frac{\sum_{\pi\in\Pi} W(\pi)\widehat{\mathrm{Reg}}_t(\pi)}{Kt\cdot\mu_t} \right),$$

can show coordinate descent returns a feasible solution after

$$\tilde{O}\left(\frac{1}{\mu_t}\right) \;=\; \tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right) \text{ steps.}$$

(Every step decreases potential by about $t \cdot \mu_t^2 = \frac{\log N}{K}$.)

Recap

Recap

Low Regret / Low Variance constraints:
implies $\tilde{O}(\sqrt{KT \log N})$ regret bound.

Recap

Low Regret / Low Variance constraints:
implies $\tilde{O}(\sqrt{KT \log N})$ regret bound.

Coordinate descent to solve LR/LV constraints:
repeatedly find a violated constraint and adjust W to satisfy it.

Recap

Low Regret / Low Variance constraints:
implies $\tilde{O}(\sqrt{KT \log N})$ regret bound.

Coordinate descent to solve LR/LV constraints:
repeatedly find a violated constraint and adjust W to satisfy it.

Coordinate descent analysis:
In round t,

$$\text{nnz}(W_t) \; = \; O(\#\text{ calls to } \arg\max_{\pi \in \Pi} \text{ oracle}) \; = \; \tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right)$$

(same as guarantee via probabilistic method).

4. Additional tricks: warm-start and epoch structure

Total complexity over all rounds

In round t, coordinate descent for computing $\boldsymbol{W}_t$ requires

$$\tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right) \text{ AMO calls.}$$

Total complexity over all rounds

In round t, coordinate descent for computing $\boldsymbol{W}_t$ requires

$$\tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right) \text{ AMO calls.}$$

To compute $\boldsymbol{W}_t$ in all rounds $t = 1, 2, \ldots, T$, need

$$\tilde{O}\left(\sqrt{\frac{K}{\log N}} T^{1.5}\right) \text{ AMO calls over } T \text{ rounds.}$$

Warm start

To compute $\boldsymbol{W}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{W}_t$.

Warm start

To compute $\boldsymbol{W}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{W}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).

Warm start

To compute $\boldsymbol{W}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{W}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).

2. Each coordinate descent step decreases potential by $\Omega\left(\frac{\log N}{K}\right)$.

Warm start

To compute $\boldsymbol{W}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{W}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).
2. Each coordinate descent step decreases potential by $\Omega\left(\frac{\log N}{K}\right)$.
3. Over all T rounds,

$$\text{total \# calls to AMO} \;\leq\; \tilde{O}\left(\sqrt{\frac{KT}{\log N}}\right)$$

Warm start

To compute $\boldsymbol{W}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{W}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).

2. Each coordinate descent step decreases potential by $\Omega\left(\frac{\log N}{K}\right)$.

3. Over all T rounds,

$$\text{total \# calls to AMO} \;\leq\; \tilde{O}\left(\sqrt{\frac{KT}{\log N}}\right)$$

But still need an AMO call to even check if $\boldsymbol{W}_t$ is feasible!

Epoch trick

Regret analysis: $\boldsymbol{W}_t$ has low instantaneous per-round regret (roughly $K\mu_t$)—this also crucially relies on i.i.d. assumption.

Epoch trick

Regret analysis: $\boldsymbol{W}_t$ has low instantaneous per-round regret (roughly $K\mu_t$)—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same W_t can be used for $O(t)$ more rounds!

Epoch trick

Regret analysis: $\boldsymbol{W}_t$ has low instantaneous per-round regret (roughly $K\mu_t$)—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same W_t can be used for $O(t)$ more rounds!

Epoch trick: split T rounds into epochs, only compute $\boldsymbol{W}_t$ at start of each epoch.

Epoch trick

Regret analysis: $\boldsymbol{W}_t$ has low instantaneous per-round regret (roughly $K\mu_t$)—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same W_t can be used for $O(t)$ more rounds!

Epoch trick: split T rounds into epochs, only compute $\boldsymbol{W}_t$ at start of each epoch.

Doubling: only update on rounds 2^1, 2^2, 2^3, 2^4, $\ldots$

Epoch trick

Regret analysis: $\boldsymbol{W}_t$ has low instantaneous per-round regret (roughly $K\mu_t$)—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same W_t can be used for $O(t)$ more rounds!

Epoch trick: split T rounds into epochs, only compute $\boldsymbol{W}_t$ at start of each epoch.

Doubling: only update on rounds $2^1, 2^2, 2^3, 2^4, \ldots$

$\log T$ epochs, so $\tilde{O}(\sqrt{KT/\log N})$ AMO calls overall.